J.1 Benchmark Instructions

J.1.1. Overview

In order to be considered for award, Contractors must successfully complete the benchmarks described below. The benchmarks may be obtained by following the instructions at http://rdhpcs.noaa.gov. Contractors that have already completed and submitted a Benchmark Software Agreement need not do so again.

The Contractor must provide in tar/gzip format the source code used and the requested verification output for all aspects of the benchmark, as described in Sections J.3.2.2.3 and J.3.2.3.3 on ISO-9660 CDROM. All written responses and spreadsheets called for in these sections must be returned with the RFP response in printed form and digitally on ISO-9660 CDROM.

J.1.2. Source Code Changes

Contractors may make changes to the compilation process and run script as necessary to accommodate their particular compilation and runtime environment(s).

Additionally, the Contractor may make changes to source code. However the Government requires that its applications be able to run on many different types of machines. Source code changes that reduce portability increase the costs of software maintenance and upgrades across multiple architectures. Therefore, certain types of code changes are preferred while others are discouraged. For the purpose of evaluating offerings, source code changes are divided into 4 Classes. The risk associated with each class of change is described below; there is no risk hierarchy implied by the use of the letters “A”, “B”, “C” or “D”.

A. Modifications required for a model to run correctly, consistent with ANSI standard FORTRAN90 and C

B. Modifications to the program parallel communication

C. Modifications consistent with ANSI standard FORTRAN90 and C

D. All other modifications

Class A modifications are those required to allow a benchmark to run to completion correctly if, without such changes to source code, the benchmark will "fail" either by exiting prior to completion or producing incorrect answers. Class A modifications do not include any changes to source solely for performance.

Since there may be many causes for such changes (e.g. existing non-standard language usage within the application, workarounds required for compiler bugs, etc), the Government cannot state categorically that such modifications will not be evaluated without some sort of risk factor assigned. Still, it is the Government's desire to consider such changes as "essentially unmodified" code with no negative impact on evaluation.

Among the types of "changes" which will be taken as Class A are:

- Use of commercially supported libraries which are bid as part of the offering and require no changes to benchmark source code or introduction of wrapper subroutines

- Compiler command lines with performance-specific options including, but not limited to, automatic parallelization

- Automatic parallelization and multitasking mediated through the operating system

- Use of commercially available and supported source pre-processors that are bid as part of the offering

Class B modifications are source code changes to the parallel communication libraries. These include the use of communication libraries other than the benchmark-provided parallel infrastructure.

Class C modifications are limited to those that do not reduce code portability and that remain consistent with ANSI standard FORTRAN90/95 and C. (It is acknowledged that the codes as they exist may already contain some ANSI non-compliant features). Performance is important and the Government is interested in performance-enhancing code modifications. However, resources to implement and maintain such changes are limited. Contractors should carefully assess and document the benefit of each specific change. Class C modifications are encouraged but as with all code changes, a risk assessment will be made.

Among the types of changes taken to be Class C are:

- Use of commercially supported libraries bid as part of the offering

- Use of compiler "directives" within the source

Class D modifications are all those changes to application source not included in Classes A, B, or C. Such modifications reduce code portability and tend to make development and maintenance more difficult and costly. Class D modifications are discouraged.

All acceptable changes must produce output that is consistent with the verification provided as described with each benchmark.

As described in the instructions below, baseline performance numbers comprised of only Class A modifications will be required. The benchmark supplied “parallel framework” will be required for this baseline where a communication library is employed. The “parallel framework” may vary from benchmark to benchmark. See the individual benchmark components for specifics. The “benchmark supplied parallel framework” is clearly not applicable for systems that use compiler or operating system mediated AUTOMATIC parallelization for the baseline benchmark.

Contractors wishing to make code changes for evaluation must submit complete performance numbers for the test suite subset affected by the code changes IN ADDITION to the baseline numbers. Having satisfied the baseline requirement, the Contractor is free to mix classes of changes. Contractors are cautioned, however, that a performance value and the set of associated changes will be evaluated as a single entity and accepted or rejected as such. The Government reserves the right to accept or reject

source code changes solely at its discretion. In the event a source code change is rejected, a detailed explanation of the reason for rejecting the change set will be provided.

J.1.3. Performance Data

Gathering of performance data is targeted for a system equivalent to that offered for the initial delivery. In this vein, the Test Systems on which the benchmarks are run and for which performance data is reported should be as close as possible to the initial offered system. In general, any component of a Test System that is not the component proposed for the initial offered system will require the Government to make a risk assessment. The reasons for assigning risk will be clearly stated to each Contractor in the Government's evaluation.

Still, given the staged delivery of resources and the significant time elapsed between proposal and delivery, the Government acknowledges that it may not be possible to use the offered system for either the RFP response or pre-delivery Live Test Demonstration (LTD). Therefore the Government will evaluate performance projection risk based on the characteristics of the test system (i.e. actual test system size, technology equivalence, etc), thoroughness of data gathering, projection methodology and vendor history. The Government is interested in understanding how and to what extent workstreams interact with each other on one or more target IT architectures comprising the Contractor's R&D HPCS System (See Section C glossary for definition of "target IT architecture"). If workstream suites can be shown to be disjoint with respect to target IT architecture, the disjoint sets may be benchmarked as separate entities.

J.1.4. NOAA Research & Development High Performance Computing System (R&D HPCS) Benchmark

J.1.4.1 Overview

The R&D HPCS Benchmark has been developed around the notion of "workstreams" (See Section C glossary for definition of "workstream").

The workstreams described below serve as a surrogate for the NOAA R&D workload and comprise the software elements of the NOAA Benchmark. Additionally, some optional standalone applications have been provided to assist the porting of benchmark components to the target platform.

Many of the workstream component applications currently run in multi-processor configurations with accompanying PE layouts. Sample PE configurations have been provided. Official RFP communication channels may be employed to request assistance with new model decomposition configurations.

The 9 NOAA benchmark workstreams are defined as:

- WS1: CM2-ESM and post-processing (Coupled Earth System Model)
- WS2: CM2-HR and post-processing (Coupled High Resolution Model)
- WS3: HIMF-VHR and post-processing (HIMF – Very High Resolution Ocean Model)
- WS4: EMTB WRF-NMM 8KM (Environmental Modeling Test Bed)
- WS5: CMDC GFS T126 (Climate Model Development and Calibration)
- WS6: DAD GSI T254 (Data Assimilation Development)
- WS7: RUC-20KM (ANX, pre-processing, forecast, and post-processing)
- WS8: WRF 5KM CHEM (WRF 5KM atmospheric chemistry)
- WS9: WRF 5KM SI (WRF 5KM static initialization)

See Section J.1.4.5, Benchmark Model Overview for more details concerning workstream components.

The R&D HPCS benchmark is comprised of 2 parts with the following goals:

i) Workstream Throughput Benchmark: A measurement of system performance under workload and Contractor-proposed runtime environment. A baseline for each workstream throughput suite is defined. The proposed R&D HPCS should minimize the execution time and maximize the overall throughput of each workstream suite targeted for a given IT architecture at a given point in time. This test defines the metric for system performance.

ii) Scaling Study: A measurement of application performance, scaling and resource requirements with respect to a given workstream component. The purpose of this test is to aid the understanding of performance projections and the intended model performance point for the offered system. The component scaling is not evaluated in and of itself. The workstream model components to be studied are:

- CM2-ESM
- CM2-HR
- HIMF-VHR
- WRF-NMM 8 KM
- GFS T126
- GSI T254

- RUC-20KM

- WRF 5KM CHEM

- WRF 5KM SI

Scaling studies are not relevant to single PE workstream elements such as the post-processing for the CM2, HIMF and RUC workstreams except as the Contractor may introduce parallelization for optimization.

Workstreams 4, 7, 8, 9 and pp6 (see Section J.1.4.5.4) are run in 32-bit, IEEE floating point precision; all other workstream components are to be run in 64-bit, IEEE floating point precision See individual workstream instructions for details.

Each throughput workstream suite baseline has been constructed from a hypothetical set of jobs that could be run on the current target IT architecture for that workstream. Multiple workstream components may run on the target IT architecture. (See Section C glossary for definition of “target IT architecture”.) For example, workstreams 1, 2 and 3 run on the SGI Origin 3900 and Altix 3700 cluster located at the NOAA, Princeton, NJ laboratory. The combined cluster produces approximately 20 “job slots” expressed in Altix 3700 processor equivalents (See Section C glossary for definition of “job slot”). These job slots have been divided between the workstreams to produce the workstream suite (i.e. the number of instances of a given workstream). It may be the case that job slots have been combined within the throughput baseline to give higher performance on the current target IT architecture.

The notion of “workstream” as it applies to workstreams 1, 2 and 3 is built around long running, self re-submitting jobs. The benchmark simulation length is based on 6 wallclock hours of runtime. The definition for “workstream” is a sequential representation for a single working set of data. From another point of view, however, the throughput benchmark attempts to create a surrogate for the system as a whole. Thus, in a given time window, multiple types of events are taking place.

For the case of workstreams 1-3, the output from completed model segments creates its own queue of work for the post-processing. Thus, the model simulation and post-processing might seem to be separate workstreams. But this view misses the fact that both input and output datasets for a single model segment are large and growing larger. Moreover, multiple model segments are used to create the datasets that are the input to the post-processing. Thus, there is a connection between input, model output and post-processing that must be preserved. This connection implies physical locality and/or very high bandwidth connectivity within a given workstream suite.

For workstreams 1-3, the connection between model output and post-processing is achieved by defining a set of file size distributions and post-processing function instance multipliers for each of the three models represented to form the workstream for that model. The file size distribution and multipliers are designed to help the Contractor understand not only how quickly the post-processing must be done, but how much post-processing there is to do on a per-workstream basis.

It is important to note that throughout the benchmark instructions, a one-to-one, though not necessarily static, mapping of application processes to physical application processors is assumed. For architectures where this is not the case, it is incumbent upon the Contractor to document the distinction between the number of application processes and application processors. In this context, "application

processors" means those processors with some part of the workstream component running on them. This does not include auxiliary processors that provide specific support functions (such as communications assists). Auxiliary processors do need to be documented as part of the system configuration.

Benchmark measurements and projections are to be reported in the spreadsheet template, Benchmark_Results.xls, which is provided with the RFP. Among other things, the spreadsheet provides for a description of the test and offered systems in terms of the processor, communication fabric and memory subsystems (see Section C for definitions). Owing to the increasing complexity of system architectures, it is not possible to provide a system description template suitably general for all cases. The Contractor should modify the system description items where necessary to describe the systems employed.

J.1.4.2 R&D HPCS Throughput Benchmark

J.1.4.2.1 General Comments

In the ideal case, throughput benchmark measurements are taken on the systems proposed for delivery using the same queuing and scheduling software being proposed for the installed system. It is understood that realization of the ideal case is highly unlikely. Thus, it is generally expected that Contractors will take performance measurements on systems with the software scheduling and queuing infrastructures currently available. After checking for interactions where the proposed R&D HPCS implies shared IT architecture components between workstream suites, projection methodologies may be used to produce the proposed configuration.

At time of delivery, Contractors will be required to demonstrate the proposed workstream suite performance for all workstreams targeted for the delivered IT architecture (i.e. it is assumed that there is a specific IT architecture within the greater R&D HPCS System for which a workstream suite has been targeted). Multiple workstream suites with a shared IT architecture target must run together in order to achieve the performance level proposed for each workstream suite. Contractors are cautioned that additional system components will be required during the contract should a workstream fail to meet the proposed performance.

It is assumed that the workstream throughput suite will be run from existing executables. Time for compilation and linking as seen by the user of the delivered system will be reported elsewhere.

The NOAA workstream throughput benchmark is comprised of:

Workstream	Number of instances
- Workstream 1: CM2-ESM + post-processing	8
- Workstream 2: CM2-HR + post-processing	6
- Workstream 3: HIMF-VHR + post-processing	4

- Workstream 4: EMTB WRF-NMM 8KM	8
- Workstream 5: CMDC GFS T126	As many as possible (see Section J.3.2.4.6)
- Workstream 6: DAD GSI T254	6
- Workstream 7: RUC-20KM forecast	6
- Workstream 8: WRF 5KM CHEM	4
- Workstream 9: WRF 5KM SI	4

J.1.4.2.2 R&D HPCS Throughput Benchmark Specific Instructions

The Contractor shall measure the workstream component timings as described for each workstream below. Based on this data and other aspects of the offered HPCS, the Contractor shall propose the Throughput Wallclock Time for each workstream. The workstream's Throughput Wallclock Time is defined as the wallclock time that elapses between the submission of the first instance of the workstream and the completion of the last instance of the workstream. For the purposes of the benchmark, it is assumed that the data in the run directory has already been staged to that run directory. At acceptance (see Section E), the Contractor shall supply and launch runscripts compatible with the offered queuing system for all workstream component jobs targeted to the proposed IT architecture. This submission time shall constitute the wallclock start time for all of the workstream components utilizing the IT architecture under test. Thus, it is essential that the Contractor account for potential interactions of workstream instances in proposing the offered throughput time for all workstreams running on shared target IT architecture components.

The runscripts used for all throughput measurements should be returned with the benchmark output.

J.1.4.2.2.1. WS1: CM2-ESM + post-processing

The model contains functions that report the Initialization, Main loop and Termination timing in terms of the minimum process time (tmin), the maximum process time (tmax) and the average process time. The Contractor should report the maximum process time (tmax) for the Initialization, Main loop and Termination for each workstream instance in Benchmark_Results.xls. The Total Throughput Time (see Section C for definition) should also be reported.

The model writes two ascii files: diag_integral.out and dynam_integral.out. These files should be returned with the benchmark output for all runs. Additionally, the model writes to stdout. This information should be captured (such as by piping to a file) and returned for all model instances. Only ascii output should be returned.

The "PP Spreadsheet" has been provided as part of the Benchmark_Results.xls. Each post-processing benchmark (i.e. pp1-9) consists of a single test function (e.g. cpio, ncks, timavg, etc). The benchmark produces a value for the elapsed time of the function for a given file size. In most cases, the elapsed time is an average time over three iterations of the function on that file size. For the baseline system (an SGI Origin 3900), the value labeled as "real" that is produced by /usr/bin/time is the wallclock elapsed time. Each of the file sizes is mapped via the "PP Spreadsheet" to a number of instances of that

function / file size combination in the 100yr post processing of the current IPCC baseline (see Section J.1.4.5.4 for more information). This mapping is done on a "model component" basis (i.e. atmosphere, land, ice and ocean). Multiplication factors are applied to each component via the spreadsheet to account for the increased file quantities and sizes of the future experiments represented by the benchmark models. Filling in the (average) elapsed time values produced by the benchmark post-processing scripts will produce a set of throughput numbers for each instance of a workstream (see the "PP Baseline" spreadsheet in Benchmark_Results.xls for an example).

There are 8 model component types:

atmos, atmos_8xdaily, atmos_level, atmos_scalar, ice, land, land_instant and ocean

that correspond to 8 post-processing time values. Each of these model component types is independent and so MAY run concurrently with any other. On the other hand, since each component takes a different length of time, the proposed architecture may "combine" some of them into sequential runs so that resources don't go idle. For example, the baseline data presented in the Benchmark_Results.xls "PP Baseline" spreadsheet shows that the IPCC ocean is currently the longest post-processing component taking over 1,000,000 wallclock seconds to complete; others such as ice and land take much less time. Therefore, one possible solution provides a percentage of resource to which the ocean post-processing is targeted while another percentage of resource handles the ice, land and any other components that can be completed as the ocean post-processing proceeds. The current set of resources for post-processing allows all components to proceed concurrently. Thus in this example, there are two "logical resource partitions" to complete an instance of the IPCC post-processing. Further, the throughput time for the IPCC baseline (and the projection to the benchmark workstreams) is simply the longest running component: the ocean. See the "PP Baseline" spreadsheet in Benchmark_Results.xls for post-processing baseline values.

For each instance of CM2-ESM, there is an instance of the post-processing. The workstream's post-processing Throughput Wallclock Time is defined as the wallclock time that elapses between the submission of the first instance of the workstream CM2-ESM and the calculated completion of the last instance of the longest running part of the workstream *post-processing* (see Benchmark_Results.xls, PP Baseline and PP Worksheet for post-processing "model component"). Given the nature of the CM2-ESM workstream, post-processing proceeds concurrently with model simulation (see Section J.1.4.1, "Overview" for the RDHPCS Throughput Benchmark). The elapsed times for post-processing tests pp1-9 are applied to the "PP Spreadsheet" included in Benchmark_Results.xls; averages over multiple test iterations are used where calculated by the post-processing scripts. The calculated elapsed time for the longest running part of the post-processing is the value placed in the "Post-processing Time for Workstream 1" field of "Benchmark Data" in Benchmark_Results.xls.

At delivery, the Contractor shall demonstrate the sustained post-processing performance proposed for the workstream. This demonstration shall be run concurrently with the portion of the HSMS benchmark (see Section J.1.4.4.2) targeted for workstream 1: 8/18 of the HSMS performance. It is assumed that all post-processing data is in "unpacked" form and resident on the "storage media" (e.g. disk array, solid state disk, etc.) utilized by the post-processing. The post-processing demonstration shall be run concurrently with all other benchmarks targeted for the CM2-ESM post-processing IT architecture. The Contractor shall run "loops" of as many instances of the post-processing sequence (pp1-9) as there are "logical resource partitions" proposed for post-processing stream multiplied by the

number of instances of workstream 1. For example, if a resource (i.e. a "logical partition") is proposed to run an instance of the ocean post-processing concurrently with another resource (i.e. another "logical partition") to run the remainder of that instance of CM2-ESM post-processing, then the eight instances of CM2-ESM imply that 16 loops of the post-processing sequence must be run concurrently across the post-processing IT architecture targeted for workstream 1. These loops shall be started at varying points in the post-processing sequence (i.e. there should be roughly equal numbers of instances started at each of the 9 items in the post-processing sequence) and the loops shall iterate for the wallclock time required to complete the HSMS performance demonstration. The Contractor shall then demonstrate that the elapsed time for each post-processing function and file size averaged over the number of iterations of the post-processing component (pp1-9) is less than or equal to the value entered as the proposed value in the "PP Spreadsheet" of Benchmark_Results.xls. In the event the HSMS benchmark requires less time to complete than a single iteration of the post-processing loop instance, additional sequential runs of the HSMS benchmark shall be made until all post-processing sequences complete. For any HSMS benchmark runs in addition to the first run, there is no requirement that the HSMS benchmark data start from tape; the first run of the HSMS benchmark is required to commence with data stored as described in Section J.1.4.4.2, "The Archive Benchmark for Workstreams 1, 2 and 3".

Like modifications to source code (see Section J.1.2), the proposed post-processing baseline must include all steps pp1-9. On the other hand, Contractors are encouraged to provide innovative solutions. Thus Contractors may propose a second set of performance numbers based on new technologies. For example, if a technology innovation removes the need for a post-processing step (e.g. through the use of archive "container files" which obsolesce the need for cpio/uncpio; see the Section C.5.2.4, "Hierarchical Storage Management System" reference to 'Storage Resource Broker'), the step may be removed from the post-processing sequence and additional sets of performance numbers proposed.

Contractors must also fill in the "Workstream Interactions" spreadsheet in Benchmark_Results.xls. This spreadsheet is intended to provide a concise, visual representation of possible job interactions arising from workstreams sharing target IT architecture components.

J.1.4.2.2.2. WS2: CM2-HR + post-processing

The instructions are identical to WS1: CM2-ESM.

J.1.4.2.2.3. WS3: HIMF-VHR + post-processing

The general instructions are identical to WS1: CM2-ESM. However, HIMF produces only timestats and stdout as useful ascii output.

J.1.4.2.2.4. WS4: EMTB WRF-NMM 8KM

The Contractor should report the wallclock time for each workstream instance of wrf.exe in Benchmark_Results.xls. The Total Throughput time (see Section C for definition) should also be reported. The Contractor should return the two ascii files standard out rsl.out.0000 and standard error rsl.error.0000 for all runs as well as the ascii file layer_statistics that can be generated using the statistics package (source is statistics.f and script is run_statistics) from the t_9000 output file. The README has a section titled STATISTICS describing in more detail how to generate the layer_statistics.

J.1.4.2.2.5. WS5: CMDC GFS-T126

The Contractor should report the wallclock time for each workstream instance of f126.64.x in Benchmark_Results.xls. The Total Throughput time (see Section C for definition) should also be reported. The Contractor should return the ascii file standard out for all runs. The Contractor should return the ascii file output generated from the rms verification program (source is rms.diff.f and script is rms.script). The rms verification program generates statistics to compare SIG forecast output files. For example, it can be used to compare the benchmark output SIG.F48 to the baseline SIG.F48. WS5 does not restrict the number of instances only that it be an even number. The WS5 forecast length may be changed via the namelist variable FHSEG. The benchmark measures the number of forecast days that can be produced in a 6-hour window with multiple streams of forecasts. See Sec J.1.4.5.6 for a more detailed description.

J.1.4.2.2.6. WS6: DAD GSI-T254

The Contractor should report the wallclock time for each workstream instance of gsi.x in Benchmark_Results.xls. The Total Throughput time (see Section C for definition) should also be reported. The Contractor should return the ascii file stdout.anl.2004010800 for all runs. The Contractor should return the ascii file output generated from the rms verification program (source is rms.diff.f and script is rms.script). The rms verification program generates statistics comparing the benchmark output signal.2004010800 to the baseline signal.2004010800.

J.1.4.2.2.7. WS7: RUC 20KM Forecast

The RUC system is composed of five (5) components: Boundary condition conversion, Analysis, pre-processing, forecast and post-processing. This benchmark represents each of these portions except the boundary condition generation portion. The boundary conditions are pre-generated for this benchmark and have been included.

Timings of each portion of this code will be returned with the benchmark output.

The clock starts when the 3Dvariational analysis begins. This portion is a single-processor code, and must be completed and post-processed before the pre-processing portion can start.

Once the analysis has been run and post-processed, the pre-processing portion can begin. The pre-processing portion combines the output from the post-processed analysis and the boundary condition files provided in order to generate the five RUC*.nnt_dat files required for the forecast engine.

As the forecast engine executes, it outputs a number of binary forecast files (e.g. yydddhh00nn.NNT_dat). These forecast output files can be post-processed (using the same post-processing executable as the one used for the analysis portion) as they become available. Processing time for each of these forecast files will likely be quite similar to one another.

The wallclock time from initialization of the analysis portion through the post-processing of the final forecast file should be reported as the workstream instance wallclock time.

A pair of sample scripts is provided. Both scripts are started; one monitors until the required data is

available and then continues with its functions. These scripts produce the required timing. One method of producing the required timing data is to port the scripts.

The Contractor should report the wallclock time for each workstream instance of RUC in Benchmark_Results.xls. The Total Throughput time (see Section C for definition) should also be reported).

J.1.4.2.2.8. WS8: WRF 5KM CHEM

The Contractor should report the wallclock time for each workstream instance of wrf.exe in Benchmark_Results.xls. The Total Throughput time (see Section C for definition) should also be reported. The Contractor should return stdout and the logfiles as the benchmark output for all runs.

J.1.4.2.2.9. WS9: WRF 5KM SI

The static initialization consists of four (4) elements gribprep, gridgen_model, hinterp and vinterp. Together, these four elements comprise the "initialization". The Contractor should report the wall clock time from the start of gribprep to the end of vinterp as the "wallclock time" for a given workstream instance of the initialization. The Contractor should report the wallclock time for each workstream instance of wrf.exe in Benchmark_Results.xls. The Total Throughput time (see Section C for definition) should also be reported. The Contractor should return stdout and the logfiles as the benchmark output for all runs.

J.1.4.2.3 R&D HPCS Throughput Benchmark Output

The Contractor shall keep the response to this section focused on the technical and engineering aspects of the benchmark data as pertains to their proposed solutions. Appropriate data includes CONCISE descriptions of the Test System configuration and any extrapolation and demonstration methodologies utilized. References to competitors or other aspects of the general computing marketplace are NOT appropriate material for this section.

The Contractor should provide a complete, concise description of the system configuration used for each of the throughput benchmark workstreams. Be sure to include:

- The number of PEs on the Test System
- Applicable PE characteristics (e.g. processor cycle time / peak performance / vector length)
- The cache configuration of each PE
- The total and application memory available to each PE
- The "communication fabric" of the system (where applicable)
- The hardware and software supporting the file system(s) for the benchmark

Provide a complete, concise description of the data-gathering procedures, the data gathered and the extrapolation methodology used. All timings are to be presented in whole units of seconds. Fractional

timings that are less than 0.5 shall be rounded “down” to the nearest integer; timings that are greater than or equal to 0.5 shall be rounded “up” to the nearest integer.

With respect to the data describing the Test System, how will the installed system differ from the Test System used for the RFP response? How do the data provided and the extrapolations from the Test System show that the installed system will perform as offered?

The file “Benchmark_Results.xls” has been distributed with the benchmark codes. In this file, an Excel spreadsheet template has been provided for the Throughput Benchmark. One spreadsheet must be completed for each of the following cases:

I. Running the Throughput Benchmark on the Test System with nothing but Class A modifications. This series of measurements constitutes the performance baseline of the offered system.

II. Running the Throughput Benchmark on the Test System with other modifications. Multiple such measurements may be provided utilizing differing modification combinations. The Contractor must make clear precisely what modifications are present to produce the measured performance and describe the mechanism of the performance enhancement. See Section J.1.2 for comments and cautions concerning use of code modifications.

III. Running the Throughput Benchmark on the Offered system with Class A modifications, if distinct from I.

IV. Running the Throughput Benchmark on the Offered system with Class A-D modifications, if distinct from II.

J.1.4.2.4 Baseline Throughput Performance

The baseline for workstreams 1, 2 and 3 was developed on a single SMP 256 processor SGI Altix 3700. The PE counts used and the number of instances are in anticipation of a system upgrade to be delivered in April, 2005. The total Altix cluster for workstreams 1, 2 and 3 should be about 3296 processors. All values for memory usage are estimated from measurements made on the SGI Origin 3000 architecture. Note that writing of history and restart files at termination uses much more memory on process 0 than per process memory utilization while executing the main loop.

Total number of processor sockets in test system:	256
Number of processor cores per socket:	001
Total number of Processor cores:	256
Processor Characteristics:	
Processor core clock speed:	1.5 GHz
Peak processor core floating point perf:	6.0 GFlops
Cache Size (L1 / L2 / L3):	32KB I&D / 256KB / 6MB
Vector Length:	Not Applicable
Total physical Memory per core:	1.0GB
Communication Fabric:	SGI NUMALink4
Supporting File System:	See C.10.1.3, “Fast Scratch”

WS1: CM2-ESM (8 instances)

Total number of processor cores:	135
Total number of MPI processes:	135
Total number of SMP threads:	N/A
Main Loop tmax (secs):	21686 secs
Initialization tmax (secs):	42 secs
Termination tmax (secs):	96 secs
Total app memory used per core:	300MB (Main loop); ~ 1.4GB at termination
Total app memory used per process:	300MB (Main loop); ~ 1.4GB at termination

Total throughput time for WS1:	21825 secs
Total post-processing time for WS1:	2,253,835 secs

WS2: CM2-HR (6 instances)

Total number of processor cores:	240
Total number of MPI processes:	240
Total number of SMP threads:	N/A
Main Loop tmax (secs):	19578 secs
Initialization tmax (secs):	854 secs
Termination tmax (secs):	568 secs
Total app memory used per core:	1GB (main loop); ~5GB at termination
Total app memory used per process:	1GB (main loop); ~5GB at termination
Total throughput time for WS2:	21001 secs
Total post-processing time for WS2:	14,199,161 secs

WS3: HIM-VHR (4 instances)

Total number of processor cores:	180
Total number of MPI processes:	180
Total number of SMP threads:	N/A
Main Loop tmax (secs):	21680 secs
Initialization tmax (secs):	54 secs
Termination tmax (secs):	73 secs
Total app memory used per core:	500MB (Main loop); ~1GB at termination
Total app memory used per process:	500MB (Main loop); ~1GB at termination
Total throughput time for WS3:	21807 secs
Total post-processing time for W3:	22,989,118 secs

The baseline for workstreams 4, 5 and 6 was developed on a 1280 processor IBM SP comprised of 40 POWER4 1.3GHZ Regatta H nodes.

Total number of processor sockets in test system: 640

Number of processor cores per socket: 002

Total number of Processor cores: 1280

Processor Characteristics:

- Processor core clock speed: 1.3 GHz
- Peak processor core floating point perf: 5.2 GFlops
- Cache Size (L1 / L2 / L3): 64KB/1MB/32MB
- Vector Length: Not Applicable
- Total physical Memory per core: 1.0GB

Communication Fabric: IBM SP Switch-2

Supporting File System: High Speed Parallel GPFS

WS4: EMTB WRF-NMM 8KM (8 instances)

- Total number of processor cores: 52
- Total number of MPI processes: 52
- Total number of SMP threads: 1
- Time (secs): 6826 secs
- Total app memory used per core: 667MB
- Total app memory used per process: 667MB
- Total throughput work/time for WS4:

    3 sequential runs (48 hour fcst) x 6826 = 20478 secs

WS5: CMDC GFS-T126 (as many instances as possible to produce the maximum number of simulation days in 6 hours of wallclock time)

Total number of processor cores:	8
Total number of MPI processes:	8
Total number of SMP threads:	1
Time (secs):	21278 secs
Total app memory used per core:	390 MB
Total app memory used per process:	390 MB

Total throughput work/time for WS5:

612 forecast hours x 54 instances = 33048 forecast hours in 21278 secs

WS6: DAD GSI-T254: (6 instances)

Total number of processor cores:	72
Total number of MPI processes:	65
Total number of SMP threads:	01
Time (secs):	1174 secs
Total app memory used per core:	513 MB
Total app memory used per process:	513 MB

Total throughput work/time for WS6:

18 sequential runs x 1174 = 21132 secs

The baseline for workstreams 7, 8 and 9 was developed on a 1528 processor Xeon Cluster with a MyriNet interconnect.

Total number of processor sockets in test system:	1528
Number of processor cores per socket:	001
Total number of Processor cores:	1528

Processor Characteristics:

Processor core clock speed:	2.2 GHz
Peak processor core floating point perf:	8.6 GFlops
Cache Size (L1 / L2 / L3):	8KB / 512KB / N/A
Vector Length:	Not Applicable
Total physical Memory per core:	1.0GB
Communication Fabric:	MyriNet
Supporting File System:	XFS

WS7: RUC-20KM forecast (6 instances)

Total number of processor cores:	100
Total number of MPI processes:	99
Total number of SMP threads:	1
3D-var Analysis	176 secs
Pre-processing (secs):	135 secs
Forecast (secs):	4200 secs
Post-processing (secs):	82
Total pre-processing memory:	512 MB
Total forecast memory used per core:	512 MB
Total forecast memory used per process:	512 MB
Total post-processing memory:	512 MB
Total throughput time for WS7:	4793 secs

WS8: WRF 5KM CHEM (4 instances)

Total number of processor cores:	64
Total number of MPI processes:	64
Total number of SMP threads:	1

Time (secs):	5700 secs
Total app memory used per core:	512 MB
Total app memory used per process:	512 MB
Total throughput time for WS8:	5700 secs

WS9: WRF 5KM SI (4 instances)

Total number of processor cores:	196
Total number of MPI processes:	196
Total number of SMP threads:	1
Time (secs):	34320 secs
Total app memory used per core:	512 MB
Total app memory used per process:	512 MB
Total throughput time for WS9:	34320 secs

J.1.4.3 R&D HPCS Scaling Study

J.1.4.3.1 General Comments

The goal of the Scaling Study is to measure individual application performance, scaling and resource requirements. Descriptions of the individual benchmark experiments are provided with each of the benchmark codes. See the README files included with the benchmark source for details. Ideally, data for the Scaling Study should be collected using the same Test System used for the Throughput Benchmark. Lacking this consistency, detailed documentation of the system differences must accompany Scaling Study data; commentary concerning the scaling and performance implications of the system differences must be provided as well.

The scaling study shall be run on all LSC architectures within the Contractor's proposed solution. Data must be returned for each architecture as described below.

Applications should be run on as few processing elements as practical for the given experiment and should be scaled to as many PEs as possible. It is clear that at some number of PEs, the performance improvement of an application with respect to a particular experiment may flatten and perhaps decline (the "rollover" point of the scaling curve). For all experiments, the Government requires data and documentation up to and including the rollover point.

J.1.4.3.2 Running the R&D HPCS Scaling Study

In order to obtain a reasonable understanding of the scaling curve, the Government requires the following minimum number of performance data points for each experiment:

WS #	Experiment	Description	Min num of data points
1	CM2-ESM	Earth System Coupled Model	6
2	CM2-HR	High Resolution Coupled Model	6
3	HIMF-VHR	Very High Resolution Ocean Model	8
4	WRF-NMM 8KM	Environmental Modeling Testbed	4
5	GFS-T126	Climate Model Dev and Cal	3
6	GSI-T254	Data Assimilation Development	4
7	RUC	Rapid Update Cycle	6
8	WRF CHEM	WRF model with atmospheric chemistry	6
9	WRF SI	5km WRF with static initialization	4

The scaling study performed on the LSC component defined as the primary target for the application has additional requirements. The Government requires that at least one of the data points be "reasonably close" (i.e. plus or minus 10%) to:

A. $1/8^{th}$ of the model application PEs proposed as the target architecture for workstream 1

B. $1/6^{th}$ of the model application PEs proposed as the target architecture for workstream 2

C. 1/2 of the model application PEs proposed as the target architecture for workstream 3

D. $1/8^{th}$ of the model application PEs proposed as the target architecture for workstream 4

E. Minimum number of application PEs proposed as the target architecture for workstream 5 for which there is sufficient physical memory to prevent paging.

F. $1/6^{th}$ of the model application PEs proposed as the target architecture for workstream 6

G. $1/8^{th}$ of the model application PEs proposed as the target architecture for workstream 7

H. $1/8^{th}$ of the model application PEs proposed as the target architecture for workstream 8

I. $1/4^{th}$ of the model application PEs proposed as the target architecture for workstream 9

Data points should be provided at reasonable intervals between the minimum number of processors used and the maximum. As an example, a requirement for “6 data points” in an experiment that needs to span the “minimum practical number of PEs” to “50% of the offered system” on a system with 1024 application PEs might look something like the set {16,32,64,128,256,512}. Contractors are encouraged to use processor configurations that take advantage of a “load balanced” number of PEs where this proves advantageous. Contractors are free to provide more data points at their discretion.

As per section J.3.1.1, Source Code Changes, the baseline measurements required of all compliant offers must be made with only Class A modifications using MPI as the message passing library for those systems employing an explicit message communication library in the benchmark. Any extrapolations of values from Test Systems to the "baseline" performance of the offered system must be based on this data.

As further described in section J.3.1.1, the Contractor may supply additional measurements and extrapolations based on any combination of Class A, B, C, or D modifications. But as noted, such a dataset is accepted and assessed risk, or rejected, as a whole. The Government will not attempt to selectively assess modifications associated with a given dataset.

J.1.4.3.2.1. CM2-ESM

The model contains functions that report the Initialization, Main loop and Termination timing in terms of the minimum process time (tmin), the maxium process time (tmax) and the average process time. The Contractor should report the maximum process time (tmax) for the Initialization, Main loop and Termination for each workstream instance in Benchmark_Results.xls.

To verify reproducibility, the Contractor should run the model for two (2) simulation days with make_exchange_reproduce=.true. for each processor configuration. The results in the files, diag_integral.out and dynam_integral.out, should be identical across these PE counts; this is a check for the atmospheric portion of the model. The reproducibility of the ocean model may be verified through a series of checksums and global integrals written to stdout at the end of the run. See the verification directory for CM2-ESM for details.

The model writes two ascii files: diag_integral.out and dynam_integral.out. These files should be returned with the benchmark output for all runs. Additionally, the model writes to stdout. This information should be captured (such as by piping to a file) and returned for all model instances. Only ascii output should be returned.

J.1.4.3.2.2. CM2-HR

The instructions are identical to WS1: CM2-ESM

J.1.4.3.2.3. HIMF-VHR

The general instructions are identical to WS1: CM2-ESM. However HIMF produces only timestats and stdout as useful ascii output. Note that HIMF-VHR does not bitwise reproduce across processor counts.

J.1.4.3.2.4. WRF-NMM 8KM

The Contractor should report the wallclock time for each configuration of wrf.exe in Benchmark_Results.xls. The Contractor should return the two ascii files standard out rsl.out.0000 and standard error rsl.error.0000 for all runs as well as the ascii file layer_statistics that can be generated using the statistics package (source is statistics.f and script is run_statistics) from the t_9000 output file. The README has a section titled STATISTICS describing in more detail how to generate the layer_statistics

J.1.4.3.2.5. GFS-T126

The Contractor should set the forecast length to 48 hours by setting the namelist variable FHSEG=48 in the execution script. The contractor is not required to provide scaling points past 63 MPI tasks.

The Contractor should report the wallclock time for each workstream instance of f126.64.x in Benchmark_Results.xls. The Total Throughput time (see Section C for definition) should also be reported. The Contractor should return the ascii file standard out for all runs. The Contractor should return the ascii file output generated from the rms verification program for the benchmark output SIG.F48 and baseline SIG.F48 (source is rms.diff.f and script is rms.script). The rms verification program generates statistics to compare SIG forecast output files.

The model should reproduce across all processor configurations.

J.1.4.3.2.6. GSI-T254

The Contractor should report the wallclock time for each configuration of gsi.x in Benchmark_Results.xls. The Contractor should return the ascii file stdout.anl.2004010800 for all runs. The Contractor should return the ascii file output generated from the rms verification program (source is rms.diff.f and script is rms.script). The rms verification program generates statistics comparing the benchmark output signal.2004010800 to the baseline signal.2004010800.

J.1.4.3.2.7. RUC 20KM Forecast

The Contractor should report the wallclock time for each configuration of the forecast engine in Benchmark_Results.xls. The Contractor should return stdout for all runs. The model should reproduce across all processor configurations.

J.1.4.3.2.8. WRF 5KM CHEM

The Contractor should report the wallclock time for each configuration of wrf.exe in Benchmark_Results.xls. The Contractor should return stdout for all runs.

J.1.4.3.2.9. WRF 5KM SI

The Contractor should report the wallclock time for each configuration of wrf.exe in Benchmark_Results.xls. The Contractor should return stdout for all runs.

J.1.4.3.3 R&D HPCS Scaling Study Output

The data to be gathered and returned with the Scaling Study benchmark are as follows:

Provide a complete, concise description of the system configuration used for the Scaling Study if different from the Test System used for the Throughput Benchmark. Be sure to include:

- Applicable PE characteristics (e.g. processor cycle time / peak performance / vector length)
- The cache configuration of each PE
- The total and application memory available to each PE
- The "communication fabric" of the system (where applicable)
- The hardware and software supporting the file system(s) for the benchmark

The Contractor should provide a complete, concise description of the data-gathering procedures, the data gathered and the extrapolation methodology used. All timings are to be presented in whole units of seconds. Fractional timings that are less than 0.5 shall be rounded "down" to the nearest integer; timings that are greater than or equal to 0.5 shall be rounded "up" to the nearest integer.

With respect to the data provided for the Test System, how will the installed system differ from the Test System used for the RFP response? How do the data provided and the extrapolations from the Test System show that the installed system will perform as offered?

The file "Benchmark_Results.xls" has been distributed with the benchmark codes. In this file, an Excel Scaling Study spreadsheet template has been provided. One spreadsheet must be completed for each of the following cases:

I. Running the Scaling Benchmark on the Test System with nothing but Class A modifications. This series of measurements constitutes the performance baseline of the offered system.

II. Running the Scaling Benchmark on the Test System with other modifications. Multiple such measurements may be provided utilizing differing modification combinations. The Contractor must make clear precisely what modifications are present to produce the measured performance and describe the mechanism of the performance enhancement. See Section J.1.2 for comments and cautions concerning use of code modifications.

III. Running the Scaling Benchmark on the Offered system with Class A modifications, if distinct from I.

IV. Running the Scaling Benchmark on the Offered system with Class A-D modifications, if distinct from II.

The Contractor should return all verification files cited in the workstream specific instructions that are produced on the Test System during the execution of the Scaling Study.

J.1.4.3.4 Baseline Scaling Performance

Scaling values for each workstream were measured on the same system as the throughput baseline. See Section J.1.4.2.4 for system configuration details.

WS1: CM2-ESM (Main loop time for 31 simulation days)

Compilation time: 5482 secs

30PE:	5360 secs (serial)
60PE:	3003 secs (serial)
90PE:	2346 secs (serial)
135PE:	1582 secs (concurrent: 90 atmosphere PEs + 45 ocean PEs)
165PE:	1468 secs (concurrent: 120 atmosphere PEs + 45 ocean PEs)
180PE:	1399 secs (concurrent: 120 atmosphere PEs + 60 ocean PEs)

WS2: CM2-HR (Main loop time for 5 simulation days; test machine had only 256 processors preventing scaling tests above 240PEs; ocean model requires 90PEs to reduce memory to 1GB per process during main loop leading to lower bound of 120PEs.)

Compilation time: 5482 secs

120PE:	4070 secs (concurrent: 30 atmosphere PEs + 90 ocean PEs)
180PE:	2542 secs (serial)
240PE:	1968 secs (serial)

WS3: HIM-VHR (Main loop time for 2 simulation days. Test system had only 256 processors.)

Compilation time: 338 secs

40PE:	4148 secs
80PE:	1869 secs
120PE:	1194 secs
160PE:	905 secs
240PE:	626 secs

WS4: WRF – NMM (Total Time 48 hour forecast)

Compilation time: 700 secs

52PE:	6826 secs
104PE:	3831 secs
156PE:	3527 secs
208PE:	2695 secs

WS5: GFS T126 (Total Time 48 hour forecast)

Compilation time: 470 secs

8 PE:	1835 secs
16 PE:	968 secs
32 PE:	510 secs

WS6: GSI - T254 (Total Time)

Compilation time: 480 secs

72PE:	1174 secs
120PE:	998 secs
168PE:	907 secs
216PE:	830 secs

WS7: RUC

Compilation time: 159 secs

36PE:	11933 secs
64PE:	7432 secs
100PE:	4793 secs

WS8: WRF 5KM CHEM

Compilation time: 587 secs

64PE: 5700 secs

120PE: 4200 secs

WS9: WRF 5KM SI

Compilation time: 3408 secs

196PE: 34320 secs

441PE: 23880 sec

J.1.4.4 HSMS ARCHIVE BENCHMARK

J.1.4.4.1 Overview

The Hierarchical Storage Management System (HSMS) archive benchmark measures the sustained throughput for moving files from the HSMS nearline tier to the files system used for the post-processing run directories. While there is no pre-award demonstration, Contractors must guarantee the run time for this benchmark that must be met at system installation.

At system installation, the archive benchmark must be run using the complete HSMS hardware and software (except for the offline tier), to confirm that the proposed run times are met by the installed system.

Contractors must fully describe the methodology by which the proposed HSMS performance is achieved. Since this benchmark will not write to archive media or move files from the post-processing run directory file system to the HSMS, the Contractor must describe all differences between read and write performance for the proposed archiving solution.

The benchmark execution time must be determined to the nearest second from “date” command output. Each file must reside on a separate tape volume and may be located at the load point of the tape. During the setup of the benchmark, Contractors must use administrator commands or other means to direct the test files to separate tape volumes. The HSMS disk cache or staging file system must be cleared before running the benchmark so that files are read entirely from tape storage. At the base period mid-life upgrade, balance of archive retrieval performance shall be maintained (see Section C.5.2.4) for all workstreams.

J.1.4.4.2 The Archive Benchmark for Workstreams 1, 2 and 3

The HSMS benchmark will measure the time to retrieve a sample dataset comprised of files representing the Government's file size distribution from the HSMS nearline tier. The sample dataset will be comprised of 252 files each approximately 5dGB in size and 648 files each approximately 450dMB in size.

The HSMS benchmark for workstreams 1, 2 and 3 must be run concurrently with the post-processing benchmarks for these workstreams (see Section J.1.4.2.2.1). File transfers may be distributed over a combination of resources if the post-processing for workstreams 1, 2 and 3 occurs on separate processing resources. The sample dataset will be retrieved from HSMS archive media to the file system used for the post-processing run directories for these workstreams.

At the initial delivery date for workstreams 1, 2 and 3, the HSMS benchmark must complete in no more than 20 minutes.

The HSMS performance is apportioned between workstreams in a weight equal to the number of workstream instances: 8/18 :: 6/18 :: 4/18 for workstreams 1, 2 and 3, respectively. The file set may be seen to divide as follows:

- 252 "large" files = 18 sets of 14 files
- 648 "small" files = 18 sets of 36 files

If the post-processing for workstreams 1, 2 and 3 occurs on separate processing resources, the file transfers will be divided among the workstreams following the 8/18 :: 6/18 :: 4/18 proportions.

J.1.4.4.3 The Archive Benchmark for Workstreams 4, 5 and 6

The archive benchmark will measure the time to retrieve a sample dataset comprised of files representing the Government's file size distribution from the HSMS nearline tier. The sample dataset will be comprised of 70 files approximately 4dGB in size and 120 files approximately 500dMB in size.

Files will be retrieved from HSMS archive media to disk storage accessible by the analysis applications for workstreams 4, 5 and 6. At the initial delivery date for workstreams 4, 5 and 6, retrieval of the sample dataset shall take less than 40 minutes.

J.1.4.4.4 The Archive Benchmark for Workstreams 7, 8 and 9

The archive benchmark will measure the time to retrieve a sample dataset comprised of files representing the Government's file size distribution from the HSMS nearline tier. The sample dataset will be comprised of 70 files approximately 4dGB in size and 120 files approximately 500dMB in size.

Files will be retrieved from HSMS archive media to disk storage accessible by the analysis applications for workstreams 7, 8 and 9. At the initial delivery date for workstreams 7, 8 and 9, retrieval of the sample dataset shall take less than 40 minutes.

J.1.4.5 Benchmark Model Overview

J.1.4.5.1 Workstream 1: CM2-ESM

The CM2 Earth System Model (ESM) is comprised of the N45L24 bgrid atmosphere core (i.e., 144 x 90 horizontal resolution with 24 levels) with land and ice model components coupled to a one-degree MOM4 ocean model. While the atmosphere portion of the model is malleable with respect to layout and PE count, the best performance of the current production model is achieved with a STATIC_MEMORY MOM4. Thus, a given executable may run multiple atmosphere configurations, but only one ocean layout. For example, the same executable may be used to run on 120 and 150 PEs in 60atm+60ocn or 90atm+60ocn concurrent mode; serial mode will always require a unique executable assuming that STATIC_MEMORY MOM4 shows performance advantages over the malleable form.

Multiple sample PE configurations have been provided. Concurrent mode examples carry the designation of the ocean portion of the model. Thus, cm2.30, cm2.60, cm2.90, cm2.120, cm2.150 and cm2.180 are all serial mode examples. Test cases cm2.30o.c (60PEs), cm2.60o.c (120 and 150PEs), cm2.72o.c (180PEs), cm2.80o.c (200PEs) and cm2.90o.c (180PEs) are all concurrent cases. All but cm2.72o.c and cm2.80o.c use executables that also run for the serial cases; the atmosphere run on 72 or 80PEs is not interesting on current architectures and so these executables have not been run in serial mode.

One of the goals of a concurrent mode configuration should be to load balance between the ocean model and remaining components. Until recently, the 60+60 and 90+90 configurations provided fairly good balance. Improvements in the time-stepping scheme for MOM4 have just been introduced that change this balance. Moreover, it's expected the port to different architectures will produce different performance features for each of the model components. Thus, finding the best balance of processing element (PE) configurations will be part of the porting task.

J.1.4.5.2 Workstream 2: CM2-HR

This model is a core benchmark application. It is comprised of the N90L40 bgrid atmosphere core (i.e., 288 x 180 horizontal resolution with 40 levels) with land- and ice-model components coupled to a 1/3rd-degree MOM4 ocean model (i.e., 1080 x 840 with 50 levels). All comments from the CM2-ESM described above apply although the PE configurations are different. Owing to the much larger model sizes than the CM2-ESM case, there are far fewer tracers and diagnostics in this workstream.

Like the ESM version, the use of the –DSTATIC_MEMORY option requires each PE to have the same number of points in each of the horizontal directions.

J.1.4.5.3 Workstream 3: HIMF-VHR

The drive to higher resolutions permeates climate research. The HIMF benchmark model is intended to be a first representative of the future classes of very-high resolution models, leading to "eddy-resolving" resolution ocean models (1/10th degree and beyond).

The HIMF model is a core benchmark. It is a 1/6th-degree hemispheric model comprised of 2160 by 680 horizontal grid points with 22 levels. This model does not use the exchange grid and thus bypasses

one of the greatest present challenges to scalability. Thus, HIMF is found to be highly scalable on current architectures and acts as a benchmark surrogate for the class of such codes.

The model is internally initialized, vastly reducing startup costs and input file size requirements. Even so, the throughput benchmark consists of running but 15 simulation days. Lower resolution test cases are provided to aid porting. There is no requirement for the Contractor to run or report performance data for any of the lower resolution cases.

Like MOM4, the use of the –DSTATIC_MEMORY option requires each PE to have the same number of points in each of the horizontal directions.

J.1.4.5.4 Post-processing for workstreams 1, 2 and 3

The post-processing benchmark components are designed to be representative of the types of operations performed for each of the workstreams 1, 2 and 3. The reporting worksheet contains multipliers to express the post-processing measurements in terms appropriate for a given workstream. The post-processing baseline is derived from the data produced by a coupled climate simulation run for the Intergovernmental Panel on Climate Change (IPCC). The data production for a single experiment running for 100 simulation years is as follows:

Each post-processing job works on one simulation year. When simulation years 5, 20, and 100 are encountered, post-processing also produces 5-year, 20-year, and 100-year averages and time series, respectively.

A single MPI process within a parallel model run produces subdomain history output in netCDF format. As the first post-processing step, the NOAA "mppnccombine" program is run to join these per-process netCDF files into a global netCDF file. The 10 GB of per-process history files produced by a one-year simulation combines to form 10 GB of global history files. The global history files for each simulation year are stored as one cpio container file in the archive.

Both the per-process and global history files and the restart files are retained in the archive. The global history files are the input to the remaining post-processing steps. After post-processing for a 100-year simulation is complete, the per-process history files are removed from the archive.

Averages and time series for each requested atmosphere, ocean, land, and ice component are produced in separate files. One TB of global history data from a 100-year simulation becomes the input to post-processing and produces 6.3 TB of output for further analysis.

A post-processing batch job works on one simulation year. Post-processing is performed for each component specified in a diagnostics table. A typical post-processing sequence yields eight components: atmos, atmos_8xdaily_instant, atmos_level, atmos_scalar, ocean, land, land_instant and ice. Further post-processing is performed as a independent batch job for each component. Yearly global history data files are stored in one 10-GB cpio container file in the archive. Each container file consists of two 6-month segments (5 GB each) of global history data. Only a subset of this input data is used, depending on what diagnostic components the user specifies. In post-processing 100 years of global history data for eight components, 6 GB of the 10 GB is typically used. Output post-processing files are again in netCDF format and contain non-time varying (“static”), climatological averages, and time series.

Climatological time averages are computed from global history data for each component. Averages may be computed on a monthly, seasonal or annual basis. Typical time averages are computed on a 1-, 5-, 20- and 100-year interval length. The batch run script will first search for output from other time averages and use that data if it is available. For example, a monthly 100-year time average will use data, if available, from monthly 20-year time averages. Lacking this, the computation will be performed on the full yearly global history data set.

Time series output data are also generated from global history data for each component. Typical frequencies for time series are 3-hour, daily, monthly, seasonal and annual.

As an example, the use of an IPCC simulation as the baseline implies the post-processing of 100 simulation years for each instance of CM2-ESM is estimated to require 2X the baseline post-processing benchmark measurements. Since post-processing on a model segment is run as a concurrent instance along with the next model segment, the model and associated post-processing run concurrently rather than sequentially to capture the essence of the workstream.

The components are constructed to run on a single PE using global history data model output in netCDF format. The components supplied as test cases are cpio/uncpio (pp1), splitvars (pp2), ncrcat (pp3), ncatted (pp4), ncks (pp5), timavg (pp6), ncap (pp7), plevel (pp8) and mppnccombine (pp9). These components make up approximately 99% of the post-processing wallclock time on the current target IT architecture: an SGI Origin3000.

Individual directories are provided for each component in the bench/run/pp directory. Each pp(n) directory contains data, scripts and output directories and a run_pp(n).csh executable script. The data directory consists of representative data of varying file sizes. The scripts directory contains c-shell, awk and Bourne shell scripts for running each component and recording the average real, user and system time and total time for the input data of several file sizes. The output directory contains a pp(n)_times.txt file containing the times recorded on the current Origin 3000 platform. The output directory also contains the stdout file from tests. Many of the netCDF operators employed produce little testable output. Use the self tests that come with the netCDF and netCDF operator (NCO) libraries to confirm functionality.

To run each component, enter the run_pp(n).csh executable. After each run, an output text file, "out", is created in each pp(n) directory. This file contains the timing information for all the input file sizes. The average real, user and system times for each file size and the total time is written to stdout.

The directories containing the source code and Makefiles or building the executables for pp2, pp6, pp8 and pp9 are in the path, bench/build/pp.

The first component, pp1, contains timing information for packing and unpacking the cpio container file of global history data. For 100 years of post-processing data for the CM2 model, cpio is executed 0 to 5 times for packing netCDF post processed data into a container file. Uncpio is executed between 3 and 22 times on the two 6-month global history files for each model component.

The utility for extracting netCDF variables from data files into individual files is splitvars (pp2). It is executed between 1 and 3 times on static and concatenated annual global history data for 100 years of post processed CM2 data. The benchmark consists of 100 KB “static data” as well as 24MB to 5GB annual global history files.

Except where noted, the post-processing benchmark data consists of annual global history files ranging from 24MB to 5GB in size.

The global history files are concatenated via the ncrcat NCO operator in the post-processing scripts. This is represented in the third test case, pp3. ncrcat is executed between 5 and 1,826 times for 100 years of CM2 post-processing.

The netCDF attribute editor, ncatted, is the fourth test case, pp4. The use of this utility is file size independent on the Origin 3000 test systems and is executed between 1 and 1,498 times for 100 years of post-processing.

The fifth test case, pp5, represents the NCO operator, ncks. This is the kitchen sink utility for extracting subsets of netCDF files and it is executed on concatenated annual global history data between 12 and 2,490 times for 100 years of post-processed CM2 data.

A component of the post-processing stream, pp6, consists of time-averaging netCDF variables of concatenated global history files. For 100 years of post processed CM2 data, timavg is executed between 5 and 2,381 times. Time averaging is executed on varying file counts, depending on the frequency of the time average. Typical time averages occur on monthly, seasonal and annual scales.

Arithmetic processing of netCDF files is carried out by the NCO operator ncap in test case pp7. It is executed only once for 100 years of post-processed data in the initial script for the first year when the netCDF attributes are copied between monthly global history data files. The benchmark files are the monthly global history files with sizes ranging from 2 to 430 MB.

The atmospheric data is processed on several atmospheric pressure levels in the test case pp8. For 100 years of post processed CM2 data, there are 17 pressure levels and plevel is executed between 5 and 19 times.

Post-processing benchmark 9 (pp9) is mppnccombine that is designed to concatenate individual process local domain history output into global domain history output.

J.1.4.5.5 Workstream 4: EMTB WRF-NMM 8KM

The benchmark for this workstream is a single forecast mesoscale model, the NMM. The NMM is only a representative of the systems run in the EMTB. Global and regional models for the atmosphere, oceans, ice, land, and space will be examined in the test bed. The model tests may be run in near real time requiring access to observation and model data from the NOAA operational HPCS (see Section C.4.4.2). The EMTB may also perform “retrospective” research from data stored in the HSMS. The projected data generated from this workstream can be found in Section C.5.2.5. Further description of this workstream can be found in Appendix A of Section C.

The NMM 8KM is a nonhydrostatic mesoscale model with NCEP dynamics and physics using the Weather Research and Forecast (WRF) infrastructure. The horizontal resolution is 8km on a central US domain with 60 vertical layers.

The executable runs on any number of PEs provided there is sufficient memory. The NMM is an MPI-only code with task counts set at the script level. It is generally not reproducible across PEs count due to a global sum. and threads. An example using 13 nodes (50 MPI compute tasks with 2 I/O tasks) run

on the NOAA IBM SP 1.3 GHz Power 4 Cluster is provided. The NMM script has a 48-hour forecast length and is controlled by the namelist variable run_days.

J.1.4.5.6 Workstream 5: CMDC GFS-T126

The benchmark for this workstream is a set of "dual runs" of the global atmospheric model, the GFS. The dual runs are a surrogate for a coupled ocean and atmosphere climate model. The ocean model has yet to be determined and is therefore represented by a second, concurrent run of the GFS. The goal is to maximize the number of forecast years. The benchmark measures the number of forecast days that can be produced in a 6-hour window with multiple streams of forecasts.

The number of instances is not specified and the Contractor is free to choose the number optimal for the offered configuration. But a "dual run" implies that the number of instances chosen must be an even number. Further, the pair of GFS instances forming the dual run must occur on a platform that would allow full communication between the pair if there were communication (as there would be if the full coupled model were running).

The projected data generated from this workstream can be found in Section C5.2.5. Further description of this workstream can be found in Appendix A of Section C.

This GFS model uses a T126 spectral resolution with 64 levels in the vertical. Documentation for GFS may be found at:

http://wwwt.emc.ncep.noaa.gov/gmb/moorthi/gam.html

The GFS executable runs on any number of PEs provided there is sufficient memory. The GFS is a hybrid MPI/OpenMP with the MPI task count and number of threads controlled at the script level. The GFS is reproducible across any number of PEs and varying numbers of MPI tasks and threads. An example using 2 nodes (8 MPI tasks with 1 thread) run on the NOAA IBM SP 1.3 GHz Power 4 Cluster is provided. The GFS example script has a 612-hour forecast length and is controlled at the script level by the namelist variable FHSEG.

Instruction for building and running the GFS are contained in the README.126.64 file in the PORT directory of the tarfile.

J.1.4.5.7 Workstream 6: DAD GSI-T254

The benchmark for this workstream is the Grid-point Statistical Interpolation (GSI). The GSI is representative of the experiments run in the Global Atmospheric Data Assimilation System. These experiments typically include atmospheric forecast models such as a high-resolution T254 GFS or the high-resolution NMM that are not included in this benchmark. The GSI can run in global and regional mode; this test only exercises the global option. The model tests may be run in near real time requiring access to the observations and model data from the NOAA operational HPCS (see Section C.4.4.2). The DAD may also perform "retrospective" research from data stored in the HSMS. The projected data generated from this workstream can be found in Section C.5.2.5. Further description of this workstream can be found in Appendix A of Section C.

The GSI model combines short-range GFS forecasts with available observations using a 3D-VAR

algorithm to produce a global analysis for subsequent GFS forecasts. The GSI differs from the NOAA Spectral Statistical Interpolation (SSI) in that it minimizes the objective function in physical (grid) space while the SSI minimizes the functional in spectral space. The GSI is slated to replace the SSI in the future.

There is no online GSI documentation available. Wu et al. (2002) discusses the GSI in a paper available from the AMS web site:

http://ams.allenpress.com/pdfserv/i1520-0493-130-12-2905.pdf

Online SSI documentation may provide useful background information. The SSI documentation is available from:

http://www.emc.ncep.noaa.gov/gmb/gdas/documentation/ssi3.html.

The GSI executable runs on any number of PEs provided there is sufficient memory. The GSI is a FORTRAN-90/95 MPI code with the MPI task count controlled at the script level. There is no threading in the GSI code. GSI results are not reproducible across varying numbers of MPI tasks.

J.1.4.5.8 Workstream 7: RUC-20KM (ANX, pre-processing, forecast, and post-processing)

The Rapid Update Cycle (RUC) is an isentropic atmospheric 4D data assimilation and forecasting system under development at the Forecast Systems Laboratory (FSL) in Boulder, Colorado and running operationally at the National Center for Environmental Prediction (NCEP). The RUC runs on an hourly cycle, utilizing its previous one-hour forecast, combined with a variety of data sources, to cycle through the next hour. The RUC currently runs operationally at a 20km resolution (112 x 151 horizontal resolution with 40 vertical levels). The RUC system also incorporates a sophisticated land-surface parameterization to represent soil and vegetation conditions. Further information can be viewed at http://ruc.fsl.noaa.gov.

The RUC is comprised of 5 components, 4 of which are represented in this benchmark. The component not included in this benchmark interpolates ETA forecasts to derive boundary conditions for the RUC. These ETA-boundary condition files have been generated and are included in the benchmark suite. The remaining subsystems are the analysis, a pre-processing step used to generate the files needed to run the forecast engine, the model forecast portion and a post-processing package that generates a variety of derived fields as well as forecasts on isobaric levels.

The analysis portion of the RUC cycle is based on a 3D variational analysis. The RUC incorporates many common data sources such as rawinsondes, surface/METAR observations, and buoy data, as well as a variety of special observations including NOAA 405 MHz and Boundary-layer profilers, RASS virtual temperatures, VAD winds, GOES precipitable water, cloud drift winds and cloud-top pressures, SSM/I and GPS precipitable water and 1500-5000 aircraft pilot reports each hour. The results of these analyses provide the initial state variables for the RUC forecasts. This portion of the RUC system currently runs on a single processor.

The RUC system must be run in a time window, that is, the analysis portion can only be started after required observations are available at about 25 minutes after the hour. Once the analysis has

completed, the model pre-processor combines the analysis output with the boundary conditions to produce the model inputs. The model can then be run, with post-processing of forecast output files being post-processed as they become available. The first of the hourly output files must have been successfully post-processed prior to the initiation of the subsequent cycle. This cycling mechanism must be demonstrated during acceptance testing (see Section E).

The pre-processing portion takes the initial conditions from the analysis as well as boundary conditions from the operational ETA forecasts at various points during the model run to produce a set of files used to drive the model forecast portion. This is essentially a pre-packaging of the data already generated.

The model is parallelized using the Scalable Modeling System (SMS), a parallel programming tool developed within NOAA. SMS provides a pre-processing system (PPP) that interprets compiler directives embedded in standard FORTRAN code to produce optimized MPI directives. This package supports both distributed and shared memory systems and has been tested on most current MPP architectures.

The model portion of the RUC requires only the 5 data files produced by the pre-processing system, an analysis time stamp (MAPTIME) and a description of the desired forecast durations and output frequencies (HYBCSTIN). Samples of these files (in big-endian format) are provided with the benchmark distribution and can be used to test the model portion separately.

These 'start-up' files can also be generated using the pre-processing software.

The final portion of the RUC system is the post-processing package. Results from the model portion are written out as IEEE binary files on the native hybrid coordinate system. The post-processing package reads these forecast files and generates a variety of derived fields as well as output on the more common isobaric coordinate system. These outputs are stored in WMO standard GRIB1 format for display and transmission.

J.1.4.5.9 Workstream 8: WRF 5KM CHEM

This benchmark utilizes the Weather Research and Forecast (WRF) under development with cooperation from multiple government and academic agencies. This version of WRF is based on the Advanced Research WRF Eulerian mass coordinate. The benchmark includes code to produce chemical tracers and incorporates cloud chemistry code to predict chemical interaction and dispersion. It should be noted that the WRF model in Workstreams 8 and 9 uses a different computational core from the WRF model used in Workstream 4.

J.1.4.5.10 Workstream 9: WRF 5KM SI

This benchmark is a test of the Weather Research and Forecast (WRF) Advanced Research version (ARW). The test contains six individual WRF tests with sample output and results for each. These six tests are: squall2d_x, squall2d_y, 3D quarter-circle shear supercell simulation, 2D flow over a bell-shaped hill, 3D baroclinic wave, and 2D gravity current. Each of these test simulations is described in the file, README_test_cases file, including an explanation of expected results.